Bilinear Fusion of Commonsense Knowledge with Attention-Based NLI Models

Amit Gajbhiye, Thomas Winterbottom, Noura Al Moubayed, and Steven Bradley

Department of Computer Science, Durham University, Durham, United Kingdom
{amit.gajbhiye, thomas.i.winterbottom, noura.al-moubayed, s.p.bradley}@durham.ac.uk

Abstract. We consider the task of incorporating real-world commonsense knowledge into deep Natural Language Inference (NLI) models. Existing external knowledge incorporation methods are limited to lexical-level knowledge and lack generalization across NLI models, datasets, and commonsense knowledge sources. To address these issues, we propose a novel NLI model-independent neural framework, BiCAM. BiCAM incorporates real-world commonsense knowledge into NLI models. Combined with convolutional feature detectors and bilinear feature fusion, BiCAM provides a conceptually simple mechanism that generalizes well. Quantitative evaluations with two state-of-the-art NLI baselines on SNLI and SciTail datasets in conjunction with ConceptNet and Aristo Tuple KGs show that BiCAM considerably improves the accuracy the incorporated NLI baselines. For example, our BiECAM model, an instance of BiCAM, on the challenging SciTail dataset, improves the accuracy of incorporated baselines by 7.0% with ConceptNet, and 8.0% with Aristo Tuple KG.

Keywords: Natural Language Inference · Commonsense Knowledge

1 Introduction

Natural Language Inference (NLI), also known as Recognizing Textual Entailment (RTE), is one of the key problems in the field of Natural Language Understanding (NLU). Popularised by a number of PASCAL RTE challenges, the task is formulated as a - "directional relationship between pairs of text expressions, denoted by T (the entailing Text) and H (the entailed "Hypothesis"). Text T, entails hypothesis H, if humans reading T would typically infer that H is most likely true." [4]. The task is very challenging as it requires an entailment system to acquire the linguistic knowledge (word meaning, syntactic structure and semantic interpretation), and also to understand commonsense knowledge.

In the context of artificial intelligence, commonsense knowledge is the set of background information about the everyday world, that an individual is expected to know or assume, and the ability to use it when appropriate [16]. Many complex NLU applications such as machine reading [21] achieved improved performance when supplied with commonsense knowledge.

Table 1. SNLI example with commonsense triples (red) from ConceptNet KG.

p: Two young girls hang tinsel on a Christmas tree in a room with blue curtains. (tinsel IsA decoration)
h: Two girls are decorating their Christmas tree. (tree RelatedTo christmas)

Thus far, NLI research has not fully leveraged the additional information available via the use of commonsense knowledge. For example, state-of-the-art NLI models [2,11] are limited to incorporating only lexical-level external knowledge, such as synonym and hypernymy. However, NLI is a complex reasoning task, in addition to lexical-level external knowledge, the task requires real-world commonsense knowledge to reason inference. Table 1 shows examples from the SNLI dataset [1], where the commonsense knowledge is retrieved from the ConceptNet Knowledge Graph (KG) [19]. The common knowledge that, *tinsel IsA decoration* and *tree RelatedTo christmas* is useful to ascertain the inference relationship. Due to the lack of such common knowledge, state-of-the-art NLI models perform substantially worse for such premise-hypothesis pairs [9].

Incorporating external commonsense knowledge in deep neural NLI models is challenging. Existing models require considerable architectural changes with marginal performance gains [2]. Incorporating such knowledge implicitly by refining word embeddings using KGs may negatively affect model performance [20]. Moreover, the existing external knowledge-based NLI models do not generalize well, and lack extensive evaluation across NLI datasets and KGs [10].

This paper aims to mitigate the aforementioned limitations. We present BiCAM (**Bi**linear fusion of **C**ommonsense knowledge with **A**ttention-based NLI **M**odels) - a novel neural network framework that incorporates NLI models without any architectural changes to the model. The BiCAM is NLI model-independent framework that generalizes across NLI models, datasets and commonsense knowledge sources. In the proposed framework, we first formulate the heuristics to retrieve commonsense knowledge from the KGs. We then embed retrieved knowledge with Holographic Embeddings (HolE) [17], a KG embedding method to learn the embeddings of entities and relations in the KG. We learn the commonsense features from KG embeddings using a Convolutional Neural Network (CNN) based encoder. Finally, we use a state-of-the-art feature fusion technique, factorized bilinear pooling, to learn the joint representation of the learned commonsense features and the sentence features from the NLI model.

Evaluation results on two established NLI baselines ESIM [3] and decomposable attention model [18], in combination with ConceptNet [19] and Aristo Tuple [5] KGs demonstrate that BiCAM considerably improve the accuracy of all the incorporated baselines. For example, compared with ESIM baseline, BiCAM achieves 7% absolute improvement with ConceptNet and 8% absolute improvement with AristoTuple KG on SciTail dataset. We analyze the effect of incorporating the different number of commonsense features and find that syntactically and semantically complex sentences require more commonsense knowledge to

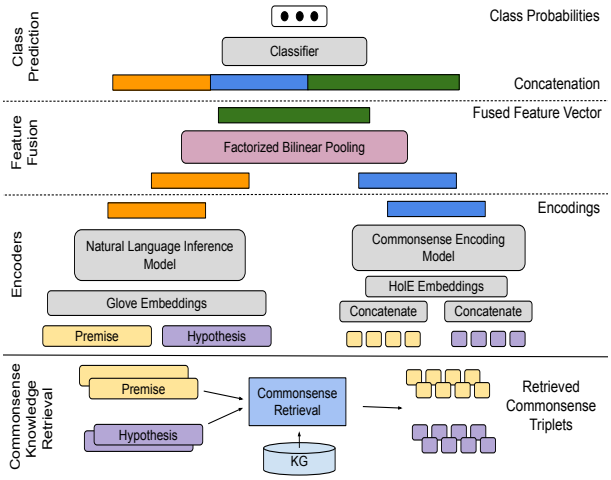

Fig. 1. A high-level view of our proposed architecture (BiCAM). The data (premise, hypothesis and the corresponding commonsense triples) flows from bottom to top. Premise and the corresponding triples are depicted in yellow, hypothesis and the corresponding triples are shown in purple.

reason inference. Further, we evaluate the impact of various feature fusion techniques and demonstrate the efficacy of bilinear feature fusion. Finally, we analyze the examples from SNLI test set, where ESIM and BiCAM succeed and fail.

In summary, the main contributions of this paper are: (1) We introduce the NLI model-independent neural framework, BiCAM, that generalizes across NLI models, datasets, and commonsense knowledge sources. (2) We devise an effective set of knowledge retrieval heuristics from KGs. (3) An extensive evaluation of the proposed approach with two established NLI baselines in combination with a general commonsense and (science) domain-specific KG on two NLI benchmarks.

2 Related Work

Leveraging commonsense knowledge in NLU systems has long been proposed [16], however, NLI neural models have only recently started utilising commonsense knowledge. KIM [2], is the state-of-the-art neural Knowledge-based Inference Model, that incorporates lexical-level semantic knowledge into the attention and composition components. Specifically, external lexical knowledge (such as synonym and antonym) extracted from the lexical database, WordNet [15], is used to form relation embeddings between premise-hypothesis words. The AdvEntuRe [11] framework train the decomposable attention model [18] with adversarial training examples generated by incorporating knowledge from linguistic resources such as WordNet, and with a sequence-to-sequence neural generator. However, lexical knowledge individually is insufficient to reason about the

premise-hypothesis relationship. Intuitively, when a human judges a premise-hypothesis relationship, a full range of real-world commonsense knowledge, and not just the lexical knowledge, is necessary to come to a conclusion [16]. Therefore, we incorporate knowledge from and empirically evaluate BiCAM on the real-world commonsense KG, ConceptNet. We also evaluate BiCAM on the (science) domain-specific KG, Aristo Tuple.

NSnet [10] is a neural-symbolic entailment model, that integrates the connectionist, deep learning approach with the symbolic approach for the scientific entailment task. The model decomposes each of the hypotheses into various facts and verifies each sub-fact against the premises using decomposable attention model and against the Aristo Tuple KB using a structured scorer. An aggregator network then combines the predictions from the two modules to get the final entailment score. Word embeddings are refined by dynamically incorporating relevant background knowledge from external knowledge sources in [20]. Our approach differs in the manner and the level at which commonsense is incorporated. We fuse the commonsense features to the sentence encodings of the premise and hypothesis which we show achieves a better performance.

3 Methods

A high-level view of our proposed BiCAM framework is illustrated in Figure 1. In this section, we discuss the individual BiCAM components and the uniquely structured framework.

3.1 Commonsense Knowledge Retrieval

To extract external commonsense knowledge we consider two KGs: ConceptNet, for general real-world commonsense knowledge and Aristo Tuple, for (science) domain-specific knowledge. The knowledge in these KGs is represented as a triple (*head*, *relation*, *tail*), where *head* and *tail* are the real-world entities and the *relation*, is a specific set of associations, describing the relation between entities. For example, (*tinsel IsA decoration*) is a triple in ConceptNet KG.

Retrieval and preparation of contextually specific and relevant information from knowledge graphs are complex and challenging tasks and is the crucial step in our model. We use a heuristic retrieval mechanism for knowledge retrieval. We find empirically that non-specific commonsense knowledge from the KGs degrades the model performance. Heuristic mechanism is fast and is effective in filtering irrelevant knowledge. We formulate the following heuristics and illustrate the triples retrieved by the application of each heuristic in Table 2.

1. Stop words are removed from the premise and hypothesis.
2. To identify the relations between the words within the premise or hypothesis, we retrieve all triples involving each pair of words as head and tail.
3. To identify the relations from premise words to hypothesis words, we retrieve the triples with premise words as head and the words of hypothesis as tail. For hypothesis, we extract the relations from the hypothesis to premise.

Table 2. A step by step illustration of commonsense knowledge retrieval for a SNLI premise-hypothesis pair from ConceptNet. Step 4 shows the final set of triplets for the premise and hypothesis.

Step	Premise	Hypothesis
Input	A **white horse** is pulling a **cart** while a **man** stands and watches.	An **animal** is walking **outside**.
1.	('white', 'horse', 'pulling', 'cart', 'man', 'stands', 'watches')	('animal', 'walking', 'outside")
2.	(horse has_property white), (cart related_to horse)	(animal at_location outside)
3.	(horse is_a animal), (horse related_to animal), (horse at_location outside)	(animal related_to horse), (animal antonym man), (animal distinct_from man)
4.	(horse has_property white), (cart related_to horse) (horse is_a animal), (horse at_location outside)	(animal at_location outside), (animal related_to horse) (animal antonym man)

4. The relation *RelatedTo* has the largest number of triples in ConceptNet. Although the relation communicates that the head and tail are related, it does not specify the specific relationship between them. To eschew the extracted commonsense knowledge from non-specific information and a higher number of triples with *RelatedTo* relation, we randomly select one triplet with *RelatedTo* relation, if multiple such triples are extracted. Additionally, we removed any duplicated triples from the final set of retrieved triples.
5. Finally, if the words of the premise and the hypothesis do not extract any commonsense knowledge by the application of above heuristics, we randomly select a word from them and extract a triple from one of the relations in ($entails$, $synonym$, $antonym$).

3.2 Encoders

Commonsense Encoding Model. The model learns the features from the retrieved commonsense triples. We provide a layer-by-layer description.

Embedding Layer. We learn the Holographic Embeddings (HolE) [17] of KG triples. Given a commonsense triple (h, r, t), HolE represents both the entities and relations as vectors in $\mathbb{R}^d$. First, HolE compose the head and tail into $\mathbf{h} \star \mathbf{t} \in \mathbb{R}^d$ using the circular correlation:

$$[\mathbf{h} \star \mathbf{t}]_i = \sum_{k=0}^{d-1} [\mathbf{h}]_k \odot [\mathbf{t}_{(k+i) \bmod d}] \tag{1}$$

where $\odot$ denotes the Hadamard product. The compositional vector obtained is then matched with the continuous representation of relation to score the commonsense triple using the scoring function defined as:

$$f_r(h,t) = \mathbf{r}^{\mathrm{T}}(\mathbf{h} \star \mathbf{t}) = \sum_{i=0}^{d-1} [\mathbf{r}]_i \sum_{k=0}^{d-1} [\mathbf{h}]_k \odot [\mathbf{t}]_{(k+i) \bmod i} \tag{2}$$

where $\mathbf{r} \in \mathbb{R}^d$ is the relation embedding. The score measures the plausibility of the commonsense triple. We train the HolE embeddings (Θ) using the pairwise

ranking loss computed as:

$$\min_{\Theta} \sum_{i \in \Gamma_+} \sum_{j \in \Gamma_-} \max(0,\ \gamma + \sigma(\eta_j) - \sigma(\eta_i)) \tag{3}$$

where Γ_+ denotes the set of triples in the KG, Γ_- denotes the "negative" triples that are not observed in KG and $\gamma > 0$ specifies the width of margin, $\sigma(.)$ denotes the logistic function and η is the value of the scoring function.

For ConceptNet and Aristo Tuple, we train the HolE embeddings for the triples retrieved from the SNLI and SciTail vocabulary. We use AdaGrad [6] to optimize the objective in Eq 3, via an extensive grid search over an initial learning rate of $(0.001, 0.01, 0.1)$, a margin of $(0.2, 1, 2, 10)$, mini-batch size $(50, 100, 150, 200)$ and entity embedding dimensions of $(50, 100, 150, 200)$. At each gradient step, we randomly generate 5 negative *tail* entities with respect to a positive triple. The learned HolE embeddings are evaluated on the triplet classification task. For SNLI/ConceptNet pair, the model achieves the highest accuracy of 64.0% with an embedding dimension of 150. For SciTail/ConceptNet and SciTail/Aristo Tuple pairs, HolE reported the top accuracy of 62.8% and 69.4% respectively at embedding dimension 100.

Encoding Layer. To learn the features over the pre-trained HolE embeddings, we employ a CNN-based neural model [13].

For each premise/hypothesis, let $\mathrm{T} = (\tau_1, \tau_2, \ldots, \tau_m)$ be a sequence of length n created by joining the m retrieved triples from the KG. Each τ is of the form (h, r, t) and, hence, $n = 3m$. The sequence T, padded where necessary, and represented as:

$$\mathrm{T} = (x_1, x_2, , x_3), (x_4, x_5, , x_6), \ldots, (x_{n-2}, x_{n-1}, x_n) \tag{4}$$

where, x_i is the i-th word in the sequence. Let $\mathbf{x}_i \in \mathbb{R}^d$ be the d-dimensional pre-trained HolE embedding corresponding to the i-th word. A sentence of length n is represented as a matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, by concatenating its word embeddings as columns, *i.e.*, $\mathbf{x}_i$ is the $i-$th column of $\mathbf{X}$. We apply a convolution operation with filter $\mathbf{W} \in \mathbb{R}^{d \times h}$, to a window of h words. The convolution operation learns a new feature map from the set of h words with the operation:

$$c = f(\mathbf{X} * \mathbf{W} + b) \in \mathbb{R}^{(\frac{n-h}{s})+1} \tag{5}$$

where, $b \in \mathbb{R}^{(\frac{n-h}{s})+1}$ is the bias term, s is the stride of convolution filter, and $f(\cdot)$ is the activation function, rectified linear unit in our experiments and $*$ denote convolution operation. The filter convolve over each window $(\mathbf{x}_{ih+1:\,(i+1)h})$ where $0 \leq i \leq n-1$ in $\mathbf{X}$. We set the h and s to 3 for the commonsense triples. Convolving the same filter with the 3-gram beginning at every 3^{rd} position in the triple sequence allows the features to be extracted from every triplet from the KG. We then apply a max-over-time pooling operation over the feature map and take the maximum value $\hat{c} = \max\{\mathbf{c}\}$ as a feature corresponding to this filter. Max pooling operation captures the most important feature for each feature map.

Above we detailed the process of extracting one feature from one filter. Multiple filters (with fixed window size and stride of 3) are employed to obtain multiple features. Each filter is considered as a linguistic feature detector that learns to recognize a specific feature from the commonsense triple. The output of the commonsense encoder is a l-dimensional vector to represent commonsense.

NLI Encoders. We incorporate BiCAM with two established NLI baselines: ESIM [3] and decomposable attention model [18].

Feature Fusion. We apply factorized bilinear pooling [22] to fuse the commonsense features and NLI sentence features. Let $\mathbf{p}$ and $\mathbf{h}$ be the NLI model generated encoding of premise and hypothesis. Also, let $\mathbf{p}_{cs}$ and $\mathbf{h}_{cs}$ denote the corresponding commonsense encoding generated by commonsense encoding model. We apply the factorized bilinear pooling defined as:

$$\begin{aligned} \mathbf{z}_p &= \text{SumPooling}(\widetilde{U}\mathbf{p} \odot \widetilde{V}\mathbf{p}_{cs}, k) \\ \mathbf{z}_h &= \text{SumPooling}(\widetilde{U}\mathbf{h} \odot \widetilde{V}\mathbf{h}_{cs}, k), \end{aligned} \tag{6}$$

where $\text{SumPooling}(x, k)$ denote a sum pooling over x with a one dimensional non-overlapped window of size k, $\widetilde{U}$ and $\widetilde{V}$ are projection matrices learned during training, $\odot$ is the Hadamard product and z is the fused feature vector. To prevent overfitting, we also added a dropout layer [8] after the element-wise multiplication of the projection matrices. Further, to allow the model to converge to a satisfactory local minimum, we append power normalization ($\mathbf{z} \leftarrow \text{sign}(\mathbf{z})|\mathbf{z}|^{0.5}$) and l_2 normalization layers ($\mathbf{z} \leftarrow \mathbf{z}/\|\mathbf{z}\|$) after SumPooling layer [22]. The factorized bilinear pooling captures the complex association between the features from premise-hypothesis and the corresponding commonsense features. The pooling method is implemented as a feed-forward neural network.

Classification Layer. We classify the relationship between premise and hypothesis using a Multilayer Perceptron (MLP) classifier. The input to the MLP is the concatenation of sentence encodings ($\mathbf{p}$ and $\mathbf{h}$) obtained from NLI model and the corresponding encodings ($\mathbf{z_p}$ and $\mathbf{z_h}$) obtained from feature fusion layer. The MLP consists of two hidden layers with *tanh* activation and a softmax output layer to obtain the probability distribution for each class. The network is trained in an end-to-end manner using multi-class cross-entropy loss.

4 Experiments and Results

Our aim is to incorporate commonsense knowledge into NLI models in order to augment the reasoning capabilities. The method should generalize across different NLI datasets, models and KGs. We evaluate BiCAM using two attention-based NLI baselines on two benchmarks in combination with two KGs. We compare our models with both external knowledge-based and attention-based NLI models. We refer to BiCAM as Bi**D**CAM, when the decomposable attention model is used as NLI baseline and Bi**E**CAM, when ESIM is used (see Figure 1).

Datasets. We assess **BiCAMs** (BiDCAM and BiECAM) on **SNLI** (570K examples) and **SciTail** (27K examples) benchmarks. We consider **ConceptNet** for general commonsense, and **Aristo Tuple** for domain-specific knowledge.

Results on SNLI. Table 3 shows the results of the state-of-the-art external knowledge-based and attention-based NLI models in comparison to BiCAMs. We evaluate ConceptNet KG for commonsense knowledge for the SNLI dataset. The models, BiDCAM and BiECAM, improve the performance of their respective attention-based baselines (decomposable attention and ESIM models) by +0.4% and +0.8%. BiCAMs also perform consistently better among the external knowledge-based and attention-based NLI models. BiECAM model achieves an accuracy of 88.8% competitive to the state-of-art external knowledge-based NLI models, ESIM+Syntactic Tree LSTM [3] and KIM [2] without any architectural changes to the underlying NLI models.

Table 3. NLI Models: Test accuracy. For our models, BiCAMs, the percentage in the parenthesis shows the performance improvement over the base models.

SNLI Dataset	
NLI Model	**Test Acc(%)**
External Knowledge-based Baselines	
AdvEntuRe [11]	84.6
BiLSTM (E_3) [20]	86.5
ESIM (E_3) [20]	87.3
Char+CoVe-L [14]	88.1
ESIM + Syntactic TreeLSTM [3]	**88.6**
KIM [2]	**88.6**
Attention-based Baselines	
CAM [7]	86.1
Decomposable Attention [18]	**86.3**
ESIM [3]	**88.0**
Our Models	
BiDCAM + ConceptNet	**86.7** (**+0.4%**)
BiECAM + ConceptNet	**88.8** (**+0.8%**)

SciTail Dataset	
NLI Model	**Test Acc%)**
External Knowledge-based Baseline	
Majority classifier [10]	60.3
AdvEntuRe(seq2seq) [11]	76.9
Attention-based Baseline	
ESIM [3]	**70.6**
Decomposable Attention [18]	**72.3**
CAM [7]	77.0
DGEM [12]	77.3
Our Models	
BiDCAM + ConceptNet	**76.8** (**+4.5%**)
BiDCAM + Aristo Tuple	**77.3** (**+5.0%**)
BiECAM + ConceptNet	**77.6** (**+7.0%**)
BiECAM + Aristo Tuple	**78.6** (**+8.0%**)

Results on SciTail. The test accuracy of different NLI models on SciTail benchmark is summarised in Table 3. For SciTail, we study the performance of BiCAMs on the general commonsense ConceptNet KG as well as the (science) domain-targeted Aristo Tuple KG. All our models significantly outperform the incorporated baselines across both the KGs, achieving absolute improvements of up to 4.5% (BiDCAM + ConceptNet), 5% (BiDCAM + Aristo Tuple) on decomposable attention baseline and 7% (BiECAM + ConceptNet), 8% (BiECAM + Aristo Tuple) on ESIM baseline. This demonstrates our framework's ability to generalize well across a number of NLI models and different KGs. All our models perform competitively on attention-based baselines, CAM and DGEM. BiECAM + Aristo Tuple observes an accuracy improvement of 1.3% over the previous state-of-the-art DGEM model.

5 Analysis

5.1 Number of Commonsense Features

To investigate the effect of incorporating various numbers of commonsense features, we vary the number of triples input to the commonsense encoding model. Particularly, we are interested in answering the question: How many commonsense features are required for optimal model performance? Figure 2 shows the results of the experiment.

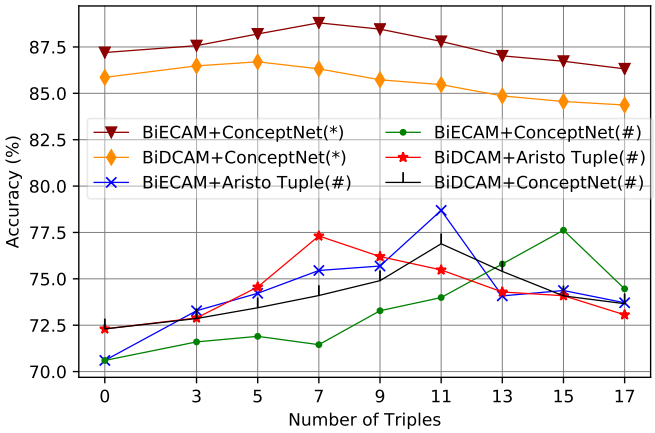

Fig. 2. Accuracy of BiCAMs with varying amount of commonsense triples. (*) denotes SNLI and (#) SciTail datasets.

For SNLI, the model BiECAM + ConceptNet achieves the highest accuracy (88.8%) using 7 triples. We observe a decrease in accuracy with increasing the number of triples. BiDCAM + ConceptNet follow the same trend, however, it attains the highest accuracy (86.7%) with the fewer number (5) of triples. The fewer number of triples required for BiCAMs to achieve their maximum accuracies on SNLI dataset, is attributed to the limited linguistic variation and short average length of stop-word filtered premise (7.35 for entails and neutral class) and hypothesis (3.61 for entails and 4.45 for neutral class) [12] of the SNLI dataset, which limit its ability to fully extract and exploit KG knowledge.

For SciTail, the BiCAMs, when evaluated using the general commonsense knowledge source ConceptNet, require a relatively high number of triplets (11 and 15 resp.) to achieve their maximum accuracy. This is due to the higher syntactic and semantic complexity of SciTail, that needs more knowledge to reason inference. However, when evaluated with the domain-specific Aristo Tuple KG, the models achieve the highest accuracies with fewer (BiDCAM at 7 and

BiECAM at 11) triples. The specialised scientific knowledge in Aristo Tuple improves the model performance with less external knowledge.

We observe that the BiCAMs, when trained on SciTail dataset, require a higher number of triples to attain maximum accuracy relative to when trained on the SNLI dataset. This can be attributed to the small training size of the SciTail dataset, which thus requires a higher number of triples to compensate for missing knowledge. We conclude that: (1) The commonsense features, when incorporated in the correct number, help reason the relationship between premise and hypothesis. (2) The number of commonsense features required depends on the syntax, semantics and size of the target dataset, as well as the domain of source KG.

5.2 Ablation Study

To evaluate the impact of factorized bilinear feature fusion, we perform an ablation study on BiECAM + Aristo Tuple, our best performing model on the SciTail dataset. Table 4 demonstrates the performance of various non-bilinear and bilinear pooling methods. We observe that factorized bilinear pooling significantly outperforms all the non-bilinear pooling methods. To ascertain that the performance gain is not due to the higher number of parameters in bilinear method, we stack fully connected layers (with 1200 units per layer, ReLU activation and dropout) to increase the parameters in non-bilinear methods. We observe that increasing the number of parameters does not increase the model accuracy. The high accuracy of factorized bilinear pooling may be attributed to the outer product between the NLI sentence and the commonsense feature vectors. Outer product allows each feature point in the two feature vectors to interact and capture associations between them. The joint representations created in such manner are more expressive than the representations created through concatenation or element-wise summation or multiplication.

For the commonsense encoder, our experiments with Recurrent Neural Networks (RNNs), LSTMs and BiLSTMs, considerably degraded the performance of the BiCAMs. This may be attributed to the inherent nature of RNNs, which learns the representations of words in the context of all previous words in the sequence. However, the set of triples input to the commonsense encoder is sequential within an individual triple. For example, in the set of triples - *(outside Antonym inside) and (table RelatedTo eating)*, the word *inside* is associated with the words in its own triple, *outside* and *Antonym*, but not with the words *table*, *RelatedTo* and *eating* of the second triple. RNNs, due to their inherent recurrent nature, learn the incorrect features from the part-sequential input of set of triples. In contrast, CNNs learns features independently of the position of words in the sequence. In the commonsense encoder, learning the features over the window of three words with a stride of three, allows the correct features to be learnt from the part-sequential set of input triples.

Table 4. Ablation Study. ($\odot$ implies Elementwise)

Fusion Method	Acc(%)
Concat	74.6
FC + Concat	75.5
FC + FC + Concat	74.3
FC + $\odot$ Sum	72.5
FC + FC + $\odot$ Sum	73.3
FC + $\odot$ Product	76.4
FC + FC + $\odot$ Product	76.8
FC + $\odot$ Difference Concat FC + $\odot$ Product	77.6
Factorized Biliniear Pooling	**78.6**

Table 5. Qualitative Analysis

BiECAM Correct ESIM Incorrect
p: Four boys are about to be **hit** by an approaching **wave**. *(wave RelatedTo crash)*
h: A giant **wave** is about to **crash** on some boys. *(crash IsA hit)*
BiECAM Incorrect ESIM Correct
p: A **red** truck is parked next to a burning **blue** building while a man in a **green** vest runs toward it. *(red Antonym blue), (blue Antonym green), (green Antonym red)*
h: The burning **blue** building smells of smoke. *(blue Antonym red), (blue Antonym green)*

5.3 Qualitative Analysis

Table 5 highlights selected sentences from the SNLI test set showing correct and incorrect inference prediction example for both BiECAM and the baseline ESIM. For the first example, BiECAM has additional context for premise and hypothesis from the knowledge that *(wave RelatedTo crash)* and *(crash IsA hit)*, which helps the model to correctly predict the inference class. However, the specific knowledge, about the *wave* and the *crash* is not available to the baseline ESIM model and hence, it incorrectly predicts the inference class.

We observe that BiECAM fails to predict the correct inference class when noisy and irrelevant knowledge is retrieved from the KGs. For example, the last test case in Table 5, only retrieves the information that colors (such as red and blue) are antonyms of each other. The retrieved knowledge is irrelevant and is not completely correct, which does not help BiECAM.

6 Conclusions

We have introduced an NLI model-independent neural framework, BiCAM, that incorporates commonsense knowledge to augment the reasoning capabilities of NLI models. Combined with convolutional feature detectors and bilinear feature fusion, BiCAM provides a conceptually simple mechanism that generalizes across NLI models, datasets and KGs. Moreover, BiCAM can be easily applied to different NLI model and KG combinations. Evaluation results show that our BiCAM considerably improves the performance of all the NLI baselines it incorporates, and does so without any architectural change to the incorporated NLI model. BiCAM achieves state-of-the-art performance on SNLI with ConceptNet KG, outperforming existing state-of-the-art external knowledge-based NLI models. Particularly for the smaller, syntactically and semantically complex

SciTail dataset, commonsense knowledge incorporation via BiCAM achieves performance improvements of 7.0% with ConceptNet and 8.0% with Aristo Tuple KG. Further analysis shows that the sufficient number of commonsense features required depends upon the syntax, semantics and size of the target dataset, as well as the domain of source KG. We observe that retrieval and selection of commonsense knowledge relevant for inference is challenging. In future work, we plan to leverage contextual word embeddings for commonsense knowledge retrieval from KGs.

References

1. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on EMNLP. pp. 632–642. ACL (2015)
2. Chen, Q., Zhu, X., Ling, Z.H., Inkpen, D., Wei, S.: Neural natural language inference models enhanced with external knowledge. In: Proceedings of the 56th Annual Meeting of the ACL (Volume 1: Long Papers). pp. 2406–2417 (2018)
3. Chen, Q., Zhu, X., Ling, Z.H., Wei, S., Jiang, H., Inkpen, D.: Enhanced lstm for natural language inference. In: Proceedings of the 55th Annual Meeting of the ACL. vol. 1, pp. 1657–1668 (2017)
4. Dagan, I., Roth, D., Sammons, M., Zanzotto, F.: Recognizing textual entailment. Morgan & Claypool Publishers (2013)
5. Dalvi, M.B., Tandon, N., Clark, P.: Domain-targeted, high precision knowledge extraction. Transactions of the ACL **5**, 233–246 (2017)
6. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. JMLR **12**(Jul), 2121–2159 (2011)
7. Gajbhiye, A., Jaf, S., Moubayed, N.A., Bradley, S., McGough, A.S.: Cam: A combined attention model for natural language inference. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 1009–1014 (Dec 2018)
8. Gajbhiye, A., Jaf, S., Moubayed, N.A., McGough, A.S., Bradley, S.: An exploration of dropout with rnns for natural language inference. In: Artificial Neural Networks and Machine Learning – ICANN 2018. pp. 157–167. Springer, Cham (2018)
9. Glockner, M., Shwartz, V., Goldberg, Y.: Breaking NLI systems with sentences that require simple lexical inferences. In: Proceedings of the 56th Annual Meeting of the ACL (Volume 2: Short Papers). pp. 650–655. ACL, Melbourne, (Jul 2018)
10. Kang, D., Khot, T., Sabharwal, A., Clark, P.: Bridging knowledge gaps in neural entailment via symbolic models. In: Proceedings of the 2018 Conference on EMNLP. pp. 4940–4945. ACL, Brussels, Belgium (Oct-Nov 2018)
11. Kang, D., Khot, T., Sabharwal, A., Hovy, E.: AdvEntuRe: Adversarial training for textual entailment with knowledge-guided examples. In: Proceedings of the 56th Annual Meeting of the ACL (Volume 1). pp. 2418–2428. Melbourne (Jul 2018)
12. Khot, T., Sabharwal, A., Clark, P.: Scitail: A textual entailment dataset from science question answering. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018 (2018)
13. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on EMNLP. pp. 1746–1751. ACL, Doha, Qatar (Oct 2014)
14. McCann, B., Bradbury, J., Xiong, C., Socher, R.: Learned in translation: Contextualized word vectors. In: Advances in NIPS. pp. 6294–6305 (2017)

15. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
16. Minsky, M.: The Society of Mind. Simon & Schuster, Inc., NY, USA (1986)
17. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 1955–1961. AAAI'16, AAAI Press (2016)
18. Parikh, A., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference. In: Proceedings of the 2016 Conference on EMNLP. pp. 2249–2255. ACL, Austin, Texas (Nov 2016)
19. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge (2017)
20. Weissenborn, D., Kočiskỳ, T., Dyer, C.: Dynamic integration of background knowledge in neural nlu systems. arXiv preprint arXiv:1706.02596 (2017)
21. Yang, B., Mitchell, T.: Leveraging knowledge bases in LSTMs for improving machine reading. In: Proceedings of the 55th Annual Meeting of the ACL. pp. 1436–1446. ACL, Vancouver, Canada (Jul 2017)
22. Yu, Z., Yu, J., Fan, J., Tao, D.: Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 1821–1830 (2017)